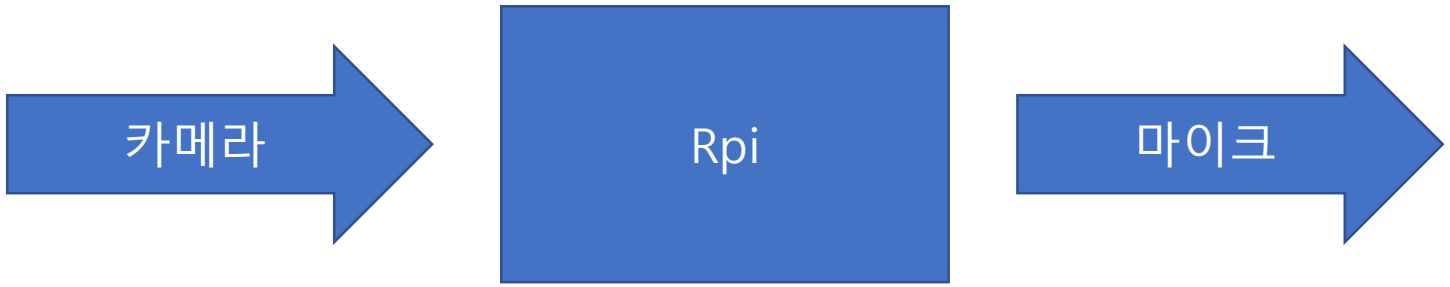
IAB – 2 얼굴인식 기반 출입자 마스크 착용 경보 시스템

전기 19 김성경

목적

"마스크를 착용하세요"

시스템 구조



구현 방식

- Opencv의 haarcascade 모델을 활용하여 얼굴을 인식
- Teachable machine을 활용해서 얼굴/마스크얼굴을 인식
- 두 모델을 겸용

시계열 데이터 처리 방식

- 실시간 영상 입력 방식이기에 3초간의 시간은 무시 가능

코드 구성(TM)

```
import tensorflow.keras
from PIL import Image, ImageOps
import numpy as np

# Disable scientific notation for clarity
np.set_printoptions(suppress=True)

# Load the model
model = tensorflow.keras.models.load_model('converted_keras/keras_model.h5')

# Create the array of the right shape to feed into the keras model
# The 'length' or number of images you can put into the array is
# determined by the first position in the shape tuple, in this case 1.
data = np.ndarray(shape=(1, 224, 224, 3), dtype=np.float32)
print(1)
# Replace this with the path to your image
image = Image.open('a23.jpg')
print(2)
#resize the image to a 224x224 with the same strategy as in TM2:
#resizing the image to be at least 224x224 and then cropping from the center
size = (224, 224)
image = ImageOps.fit(image, size, Image.ANTIALIAS)

#turn the image into a numpy array
image_array = np.asarray(image)

# display the resized image
image.show()

# Normalize the image
normalized_image_array = (image_array.astype(np.float32) / 127.0) - 1

# Load the image into the array
data[0] = normalized_image_array

# run the inference
prediction = model.predict(data)
print(prediction)
```

```
import numpy as np
import cv2
import os
import tensorflow.keras
from PIL import Image, ImageOps
xml = 'haarcascade_frontalface_default.xml'
face_cascade = cv2.CascadeClassifier(xml)
f = "face.mp4"

cap = cv2.VideoCapture(0) # 노트북 웹캠을 카메라로 사용

#cap.set(3,640) # 너비

#cap.set(4,480) # 높이
#model = tensorflow.keras.models.load_model('keras_model.h5',compile=False)

def classify(image):
    # Disable scientific notation for clarity
    np.set_printoptions(suppress=True)

    # Load the model
    model = tensorflow.keras.models.load_model('keras_model.h5',compile=False)
```

```
#print(model)

data = np.ndarray(shape=(1, 224, 224, 3), dtype=np.float32)
size = (224, 224)
image = ImageOps.fit(image, size, Image.ANTIALIAS)
#turn the image into a numpy array
image_array = np.asarray(image)
# display the resized image
#image.show()
normalized_image_array = (image_array.astype(np.float32) / 127.0) - 1

data[0] = normalized_image_array

# run the inference
prediction = model.predict(data)
print(prediction)
prediction=prediction[0]
if prediction[0]>=prediction[1] and prediction[0]>=prediction[2]:
  print("no face")
  return 0
if prediction[1]>=prediction[2] and prediction[1]>=prediction[0]:
  print("pure face")
```

```
    if prediction[2]>=prediction[1] and prediction[2]>=prediction[0]:
      print("mask face")
      return 2


framecnt = 0
while(True):
    framecnt+=1
    print("onframe"+str(framecnt),end="")
    ret, frame = cap.read()

    if framecnt%10==0:

        frame = cv2.flip(frame, 1) # 좌우 대칭
        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

        faces = face_cascade.detectMultiScale(gray)
        print("Number of faces detected: " + str(len(faces)))
        retp = 0
        if len(faces):
            for (x,y,w,h) in faces:
                cv2.rectangle(frame,(x,y),(x+w,y+h),(255,0,0),2)
                if w*h>=5000:
                    retp+=1
        print("Number of REAL faces detected: " + str(retp))
        if retp>=1:
            file = "helloEN.mp3"
            os.system("mpg123 " + file)

        cv2.imshow('result', frame)
        #resu=classify(frame)
        #print(resu)
```

```
        k = cv2.waitKey(30) & 0xff
        if k == 27: # Esc 키를 누르면 종료
            break

cap.release()
cv2.destroyAllWindows()
```

시연